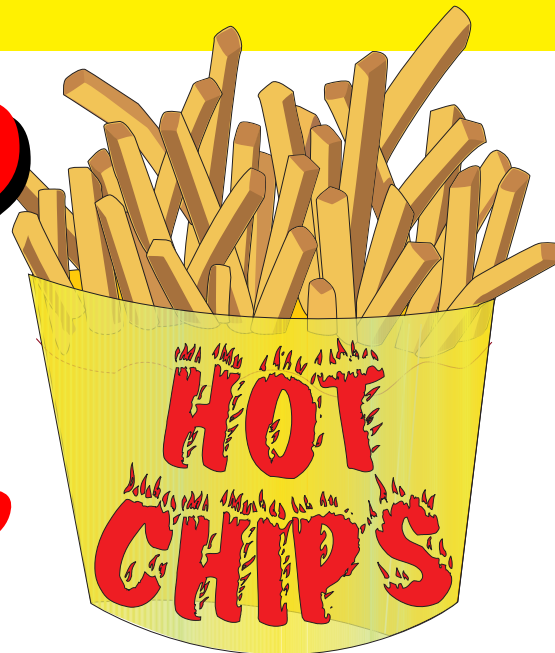


HOT CHIP?



Do you want a drink with that?

**Are you currently learning about microcontrollers?
Thinking about a project that has real potential?
Need to do something more than flash a LED or sound a buzzer?
The “Hot Chip Starter Kit” is worth a close look.**

This new micro kit from Quasar Electronics will interest both the beginner and expert alike. Beginners will find that they can write and test a simple program, using the BASIC programming language, within an hour or two of connecting it up. Experts will like the power and flexibility of the Atmel microcontroller, as well as the ease with which the little Hot Chip board can be “designed-in” to a project.

What’s in the box?

The Hot Chip Starter Kit includes just about everything you need to get up and experimenting right away. On the hardware side, there’s a pre-assembled microcontroller PC board with both parallel and serial cables for connection to your PC. Software on CD-ROM for Windows 3.1, Windows 95 and Windows 98 is included, and features an Assembler, BASIC compiler and in-system programmer.

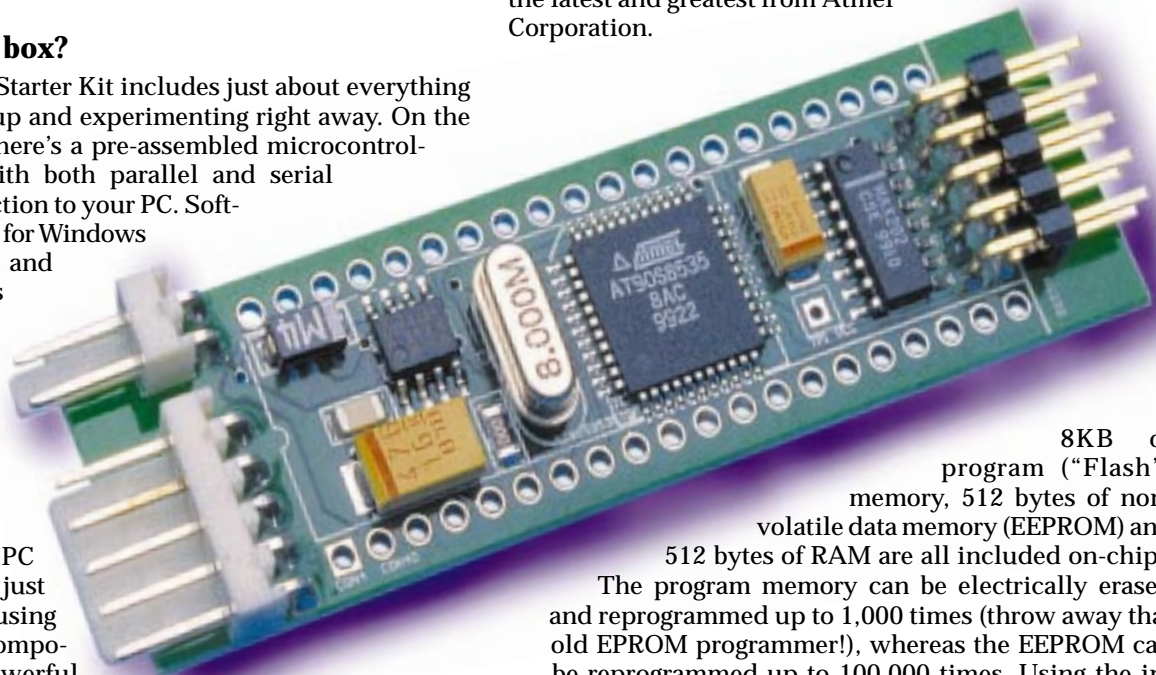
Hardware

The Hot Chip PC board measures just 20 x 70mm but using surface mount components and a powerful

microcontroller, it packs an incredible amount of functions into a small space.

What is a microcontroller? A microcontroller integrates a microprocessor core with key peripherals such as RAM, ROM, I/O ports, counter/timers, serial ports, A-D converters, etc – all on a single chip.

The kit uses the AT90S8535 microcontroller IC, one of the latest and greatest from Atmel Corporation.



8KB of program (“Flash”) memory, 512 bytes of non-volatile data memory (EEPROM) and 512 bytes of RAM are all included on-chip.

The program memory can be electrically erased and reprogrammed up to 1,000 times (throw away that old EPROM programmer!), whereas the EEPROM can be reprogrammed up to 100,000 times. Using the in-

"Hot Chip" Microcontroller Starter Kit from DSE

system programming (ISP) features of the microcontroller, the Hot Chip software can erase and reprogram both program and data memory via your PC's parallel port, all in a matter of seconds.

Table 1 lists all the major features of this little powerhouse. Further information can be downloaded from the Atmel web site at www.atmel.com – look under the "AVR 8-bit RISC" Microcontrollers section. (As a matter of interest, we used a similar, though smaller, Atmel microcontroller IC back in the November 1999 issue to control our LED Christmas Tree project).

So far we've only talked about the microcontroller chip itself, but the Hot Chip PC board includes a number of other components to make it easier to use "out of the box".

DC power between 9 and 12V is supplied to the board via a 2-pin connector. This supply is regulated and filtered to 5V. The kit includes a power cable for connection to a 9V battery (it draws only 30mA), but any DC supply within the specified range could be used. A series diode provides

polarity protection at the input.

Power-on reset and brownout (low voltage) protection is provided for the microcontroller and a 32kHz crystal has been included to make it easy to set up a "real time" clock.

A Maxim RS-232 IC converts the microcontroller's serial port signals to RS232 levels, which are then made available on a 10-pin connector. A cable is supplied with the kit for connection to COM1, 2, 3 or 4 on your PC. (Your computer only has COM 1 and 2 and both are used? Have a look at your PC's instruction manual – you'll almost certainly find that COM 3 and 4 are also available.)

As mentioned above, programming the microcontroller's memory is performed via your PC's parallel (LPT) port. All hardware support for this feature is provided within the microcontroller chip itself, with the necessary signals brought out to a 5-pin connector via current-limiting resistors. A cable is supplied with the kit for connection to either LPT1 or LPT2 on your PC. Similarly to COM ports, many computers only have LPT1 brought to an

Table 1: AT90S8535 Microcontroller Features

- **AVR® - High-performance and Low-power RISC Architecture**
 - 118 powerful instructions – most single clock cycle execution
 - 32 x 8-bit general purpose working registers
 - Up to 8 MIPS (Millions of Instructions Per Second) throughput at 8 MHz
- **Data and Non-volatile Program Memories**
 - 8K Bytes of in-system programmable flash memory SPI serial interface for in-system programming endurance: 1,000 Write/Erase Cycles
 - 512 Bytes EEPROM endurance: 100,000 write/erase cycles
 - 512 Bytes internal SRAM
 - Programming lock for software security
- **Peripheral Features**
 - 8-channel, 10-bit A-D converter
 - Programmable UART (Universal Asynchronous Receiver and Transmitter) (Serial Port)
 - Two 8-bit timer/counters with separate prescaler and compare mode
 - One 16-bit timer/counter with separate prescaler, compare and capture modes and dual 8-bit, 9-bit, or 10-bit PWM
 - Programmable watchdog timer with on-chip oscillator
 - On-chip analog comparator
- **Special Microcontroller Features**
 - Power-on reset circuit
 - Real Time Clock (RTC) with separate oscillator and counter mode
 - External and internal interrupt sources
 - Three sleep modes: idle, power save, and power down
- **Power Consumption at 4 MHz, 3V, 20°C**
 - Active: 6.4 mA
 - Idle mode: 1.9 mA
 - Power down mode: <1µA
- **I/O and Packages**
 - 32 Programmable I/O lines
 - 40-pin PDIP, 44-pin PLCC and 44-pin TQFP (*Hot Chip uses 44-pin TQFP package*)
- **Operating Voltages**
 - V_{CC}: 4.0 - 6.0V AT90S8535 (*Hot Chip uses +5.0V*)
 - V_{CC}: 2.7 - 6.0V AT90LS8535
- **Speed Grades:**
 - 0-8MHz AT90S8535 (*Hot Chip speed is 8MHz*)
 - 0-4MHz AT90LS8535



What's in the box? The Hot Chip Starter Kit includes just about everything you need to get up and experimenting right away: a pre-assembled microcontroller PC board with both parallel and serial cables for connection to your PC. There's also software on CD-ROM for Windows 3.1, Windows 95 and Windows 98 which features an Assembler, BASIC compiler and in-system programmer.

wouldn't recommend extending the parallel cable as programming errors may result.

Design-in

The Hot Chip PC board has two rows of 20 pads that provide convenient access to all microcontroller pins. It's no accident that these pads have the same spacing as a 40-pin IC! If two 20-way SIL (single in-line) pin headers are installed, the board can be plugged into a 40-pin IC socket as part of a larger project. Does it get any easier than this?

Alternatively, if you want to use the board in "stand-alone" mode, you can do that too. Individual pins or stakes can be soldered into whichever pads you desire – loose pins are provided with the kit.

The Hot Chip on-line documentation includes a circuit diagram and PCB layout that can be printed for ease of reference.

Software

Software for Windows 3.1x, 95 and 98 is provided on CD-ROM, along with a complete technical manual for the AT90S8535 microcontroller in Windows help file format.

The software consists of several major components, all of which are accessible from a simple to use graphical interface called "Debug ABC" (Fig. 1).

Programs are entered using any text editor (Fig.3). Windows Notepad is preferable to using your normal word processor: most word processing programs can be a trap for young players, especially if you forget to save in text-only mode. Most word processors embed codes in your text which are invisible to you – but not the software!

By the way, if you don't already have a good text editor, we suggest giving Programmers File Editor (PFE) a try. It is available for free download from www.simtel.net/pub/

outside socket at time of manufacture (to save a few cents!). As most printers are hooked up to LPT1, you'll find it much more convenient to use LPT2 if it is available. Again, check your PC operating manual.

The parallel cable supplied is only 80cm in length and the serial cable is even shorter, so you'll need to position the Hot Chip PC board right next to your PC. Although you could easily extend the serial cable without problems (simply use an appropriate male/female serial cable), we



Fig.1: all software functions are accessible from the main window.

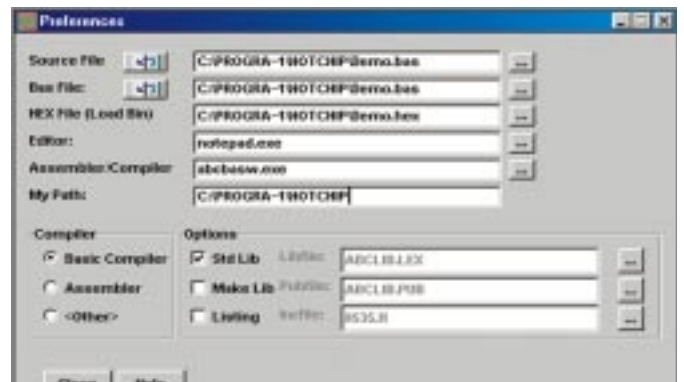


Fig.2: likewise, all preferences are easily set from a sub-window.

simtelnet/win95/editor/pfe101i.zip

Programs can be written in BASIC or Assembly language (or both). Before BASIC programs are ready to run they must go through a two-step process. First, the BASIC compiler converts (compiles) the program into lower-level (but functionally equivalent) instructions in Assembly language (Fig.4).

Then the Assembler translates these instructions into machine code (binary) format (Fig.5), ready for programming into the microcontroller's Flash memory.

If you're familiar with BASIC programming, you'll find most of the syntax quite familiar. Even if you're not, one of the best things about BASIC is that you will have your first program up and running in quick time.

Note that although compiled BASIC programs are notorious for their slow execution speeds, the Atmel AVR series of microcontrollers are specifically tuned for running compiled code ('C' in particular but we can't see why this wouldn't apply to BASIC as well).

So unless you have a time-critical application that requires microsecond accuracy, you will probably find that BASIC does the job just fine.

Once you have a program that you are ready to commit to memory, it's simply a matter of clicking the Erase button, then the Program button to write it to the microcontroller's Flash memory (Fig.1). Support is also provided to enable reading and writing of EEPROM memory, either as individual bytes or from data stored in a file.

Once programming is complete, execution begins when you click on the *UNreset* button, releasing the microcontroller's reset line.

If your program reads and/or writes to the microcontroller's serial port (UART), you can 'talk' to it via the serial communications module (Fig.6). Once again, this is accessible from the main (Debug ABC) window. This is of course the purpose of the serial cable connecting the Hot Chip board to the PC's COM port.

Summary

The only negative comment about the kit is that the software lacks any kind of real debugging tools or support for industry-standard tools such as those found in AVR Studio. If you need to do any serious debugging, you're on your own...

Nevertheless, we think that the low cost, power and versatility of the Hot Chip Starter Kit makes it an excellent choice for students, hobbyists and professionals. SC

Where do you get it?

The Hot Chip Starter Kit is available through Quasar Electronics Limited by mail order.

TEL: 0870 246 1826

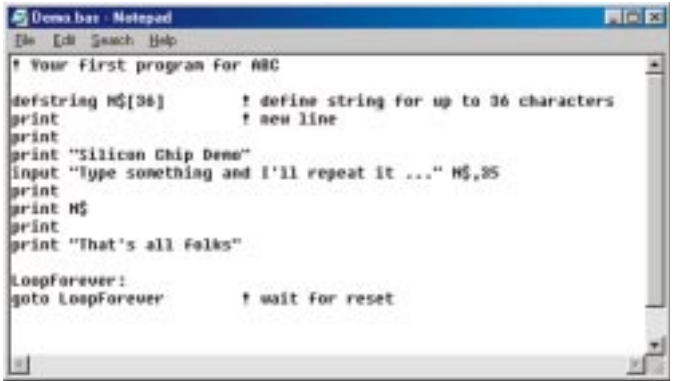
FAX: 0870 460 1045

EMAIL: sales@QuasarElectronics.com

Retail price is £49.95 inc. VAT (plus p&p)

You can find full product details on the Quasar Electronics website at:

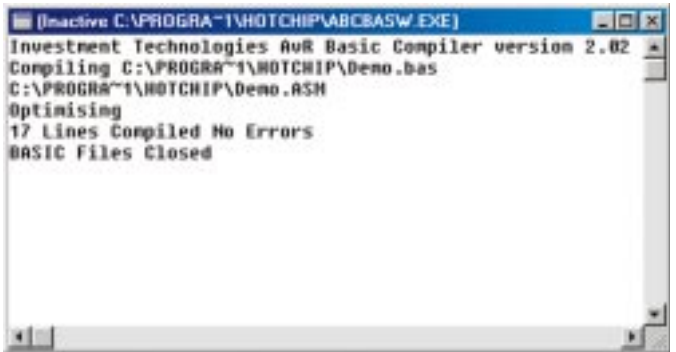
www.QuasarElectronics.com/abc_products.htm



```
! Your first program for ABC
defstring M$(36)      ! define string for up to 36 characters
print                ! new line
print
print "Silicon Chip Demo"
input "Type something and I'll repeat it ..." M$,35
print M$
print
print "That's all folks"

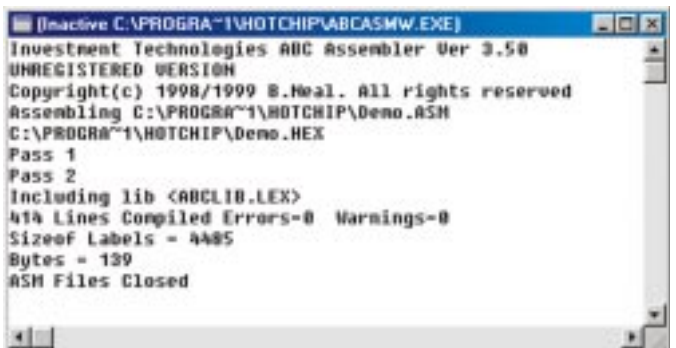
LoopForever:
goto LoopForever      ! wait for reset
```

Fig.3: both BASIC and Assembler programs are entered using your favourite text editor. The default editor is Windows Notepad but this can be changed in the Preferences dialog box.



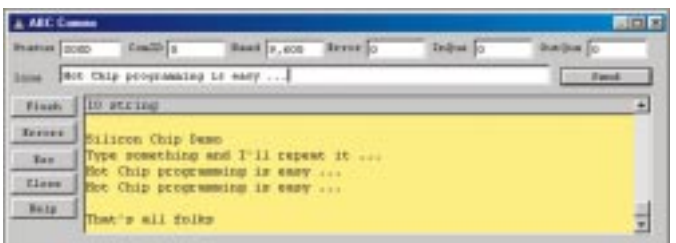
```
Investment Technologies AVR Basic Compiler version 2.02
Compiling C:\PROGRAM\HOTCHIP\Demo.bas
C:\PROGRAM\HOTCHIP\Demo.ASH
Optimising
17 Lines Compiled No Errors
BASIC Files Closed
```

Fig.4: clicking on the *Assemble* button launches the Compiler. Any errors that are detected are displayed along with the line number that generated them.



```
Investment Technologies AVR Assembler Ver 3.50
UNREGISTERED VERSION
Copyright(c) 1998/1999 B.Neal. All rights reserved
Assembling C:\PROGRAM\HOTCHIP\Demo.ASH
C:\PROGRAM\HOTCHIP\Demo.HEX
Pass 1
Pass 2
Including lib <ABCLIB.LEX>
474 Lines Compiled Errors=0 Warnings=0
Sizeof Labels = 4485
Bytes = 139
ASH Files Closed
```

Fig.5: the output from the compiler is automatically passed to the Assembler, which produces the binary file ready for programming into the microcontroller.



```
Hot Chip programming is easy ...
Flash: 100 string
Erase: Silicon Chip Demo
Go: Type something and I'll repeat it ...
Flash: Hot Chip programming is easy ...
Erase: Hot Chip programming is easy ...
Go: That's all folks
```

Fig.6: the simple demo program we wrote in Fig.3, compiled in Fig.4 and assembled in Fig.5 is now "talking" to the PC via the COM1 port.